UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/059,427	01/29/2002	Jeroen Anton Johan Leijten	NL010073	6839

24737 7590 05/04/2007

PHILIPS INTELLECTUAL PROPERTY & STANDARDS
P.O. BOX 3001
BRIARCLIFF MANOR, NY 10510

EXAMINER	
PETRANEK, JACOB ANDREW	

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
05/04/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/059,427
Filing Date: January 29, 2002
Appellant(s): LEIJTEN, JEROEN ANTON JOHAN

Dicran Halajian, Reg. No. 39,703
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 11/13/2006 appealing from the Office action mailed 9/22/2005.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

A substantially correct copy of appealed claim 4 appears on page 16 of the Appendix to the appellant's brief. The minor errors are as follows: The status identifier of claim 4 should be previously presented since amendments have been made to the claim. Also, a period is missing from the claim that was added during the last amendment to the claims.

(8) Evidence Relied Upon

Miller et al. (U.S. 5,819,058), Mohamed et al. (U.S. 6,684,319), and Keller et al. (U.S. 6,546,478) are relied upon as evidence.

(9) Grounds of Rejection

The current examiner working on the case has added clarification to claim 1 and made minor changes to the rest of the rejection that a previous examiner made.

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1, 4, and 6-8 are rejected under 35 U.S.C. 102(b) as being anticipated by Miller et al. (U.S. 5,819,058).

3. As per claim 1:

Miller disclosed a computer system with processing unit being arranged to fetch memory lines from the memory and execute instructions from the memory lines (Miller: Figure 5 elements 26, 30-36, and 172, column 8 lines 11-29 and lines 41-62)

Each memory line being fetched as a whole and being capable of holding more than one instruction (Miller: Figures 7 and 8, column 8 lines 11-29 and column 11 lines 58-61)(The first line in figure 7 shows VLIW packets IP0, IP1, IP2, and IP3, which results in a single memory line containing more than a single instruction. Column 8

states that 128 bits can be fetched at once, which is a memory line. Figure 8 shows an example of an entire memory line being fetched.)

At least one instruction comprising information, inserted at compile time (Miller: Figure 3b element 117, column 6 lines 31-35 and lines 53-58)(Figure 3b is a compressed 32-bit instruction. The compiler performs the compression of the VLIW packets and inserts element 117 into the instructions to tell if the current instruction is at the end of a memory line or not. Element 117 reads upon information.),

That signals explicitly how the processing unit, when processing the instruction from a current memory line, should control how a part of processing is affected by crossing of a boundary to a subsequent memory line (Miller: Figures 3b, 7, and 8 element 117, column 6 lines 31-35 and column 11 lines 58-67 continued to column 12 lines 1-25)(Column 6 describes a 32-bit compressed instruction containing 117 that tells if the current instruction is the last instruction within the VLIW packet. Figures 7 and 8 show examples of fetching VLIW packets from memory. IP3 and IP4 are examples of VLIW packets crossing into the subsequent memory line. Looking at figure 8, IP3 is located in memory segments 6-9 in memory lines 0 and 1. It's assumed that IP3 contains two 32-bit compressed instructions for this example. The 32-bit compressed instruction in memory segments 6-7 contains element 117 signaling not-end-of-packet since another compressed 32-bit instruction is in memory segments 8-9. Thus, element 117 will indicate that the subsequent memory line must be read out of the memory to get the whole VLIW packet to completely decompress the VLIW packet.),

The processing unit being arranged to respond to the information by controlling said part as signaled by the information (Miller: Figures 7 and 8, column 11 lines 58-67 continued to column 12 lines 1-25)(IP3 and IP4 have to fetch from the next memory line due to the instruction crossing the memory boundary.).

4. As per claim 4:

Miller disclosed a computer system according to claim 1, wherein information signals explicitly whether or not an instruction pointer should be updated from a position behind the instruction in the current memory line to a start of the subsequent memory line, so that information following the instruction on the current memory line is skipped over, the processing unit being arranged to update the instruction pointer to the start of the subsequent memory line in response to the information (Column 11, lines 35-45, Column 12, lines 10-25 and figures 7 and 8; Miller teaches that the addressing system determines that a next instruction packet has a pad instruction in front of it. It is inherent that there is information that explicitly signals the addressing system in order for the addressing system to determine there is a pad instruction, a determination couldn't be made by hardware without an explicit signal. The pad instruction is then discarded and does not cause any operation to occur in the processor, and the Aright and Aleft addresses (program counters) are incremented to point to a new memory line, therefore the pad instruction is skipped over. The Aright address points to byte 12 in memory 282 (a subsequent memory line) and Aleft points to byte 16 (a subsequent memory line) of memory 280.)

5. As per claim 6:

Miller disclosed a computer system according to claim 1, the processing unit being a VLIW processing unit containing two or more issue slots for issuing operations from the instruction in parallel to the functional units: [Figs. 5, 6 and 9. Col. 2, lines 27-63]

-The instructions being VLIW instructions, capable of containing two or more operations: [Figs. 5, 6 and 9. Col. 2, lines 27-63]

-The instruction comprising a field distinct from the operations to specify said information: [Fig. 4, Token field 134, col. 5, lines 39-53.]

6. As per claim 7:

Miller disclosed a computer system according to claim 6, the field comprising, in addition to said information, a decompression code that specifies for which issue slots the instruction contains operations: [Col. 5, lines 39-53, figs. 5 & 6. The token indicates the instruction type and "identifies the processing unit."]

7. As per claim 8:

Claim 8 essentially recites the same limitations of claim 1. Therefore, claim 8 is rejected for the same reasons as claim 1.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 2, 3, and 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Miller et al. (U.S. 5,819,058), in view of Mohamed et al. (U.S. 6,684,319).

10. As per claim 2:

Miller disclosed a computer system according to claim 1.

Miller failed to teach selectively loading prefetched instructions based on the information in the token field.

However, Mohamed disclosed wherein information signals explicitly whether or not the subsequent memory line has to be prefetched during processing of VLIW instructions, the processing unit being arranged to start prefetching of the subsequent memory line in response to the information: (Column 3, lines 35-60)

It would have been obvious to combine the prefetch instruction flag bit of Mohamed with the processor of Miller because of the advantages explained in Mohamed in col. 2, lines 35-64, including decreasing the power consumed that is normally associated with prefetching and the time required when fetching long instructions. The prefetching of long instruction words (128 or 256 bits for example) can cause a significant amount of power to be consumed (column 1, lines 21-32 of Mohamed). Therefore, adding an additional bit for selective prefetching as taught by Mohamed would have saved power, because not every line would need to be prefetched, only those that are necessary. This advantage of using less power would have provided the motivation to add the prefetch instruction flag bit.

11. As per claim 3:

Miller and Mohamed disclosed a computer system according to claim 2, wherein the information contains a prefetch bit whose value signals explicitly whether or not the subsequent memory line has to be prefetched: (The combination of Miller and Mohamed as applied to claim 2 teaches the limitations of claim 3. See column 3, lines 35-60 of Mohamed)

12. As per claim 9:

The additional limitation(s) of claim 9 basically recite the additional limitation(s) of claim 2. Therefore, claim 9 is rejected for the same reason(s) as claim 2. Examiner notes that the claim language requires "at least one of" a list of things, therefore this combination of Miller and Mohamed is sufficient to reject claim 9.

13. Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Miller et al. (U.S. 5,819,058), in view of Keller et al. (U.S. 6,546,478).

14. As per claims 5:

Miller disclosed a computer system according to claim 1.

Miller failed to teach wherein the information signals explicitly whether or not processing of the instruction should be stalled, when the instruction is reached from a branching instruction, processing being stalled to fetch the subsequent memory line that contains a part of the instruction, the processing unit being arranged to stall in response to the information when the instruction is reached from the branching instruction.

However, Keller disclosed information that signals (Continuation field 126 of figure 8) explicitly whether or not processing of the instruction should be stalled, when

the instruction is reached from a branching instruction, processing being stalled to fetch the subsequent memory line that contains a part of the instruction, the processing unit being arranged to stall in response to the information when the instruction is reached from the branching instruction: (Column 22, lines 1-17, column 23, lines 17-24 and column 26, line 44 to column 27, line 8; The invention of Keller uses the same PC address used to fetch instructions from the I-cache to fetch a line predictor entry 82 of figure 6 (PC shown in fig. 3). This entry contains information, including the continuation bit 126 (figure 8), which is combined with the instruction to be decoded by the decoder (fig. 4, fig. 22).

This information, while associated with the instruction and addressed by the same address as the instruction, is stored in a different memory than the instruction. It would also have been obvious to combine the continuation bit 126 in the instruction itself instead of in a separate line predictor entry 82 since it has been held that the use of a one piece construction (instead of the two piece line predictor entry structure and I-cache disclosed in Keller) "would be merely a matter of obvious engineering choice." *In re Larson, 340 F.2d 965,968, 144 USPQ 347,349 (CCPA 1965).*

Adding the continuation bit to the instruction information bits that already exist would allow the processor to know whether a branch target instruction crosses a boundary in the cache and a stall should occur to fetch the next cache line. The processor could detect the need for a memory access quickly and easily if the continuation bit was included in the cache line. The earlier a memory access is known to be needed the better, because it is well known in the art that memory accesses can

be costly to processing speed. Specifically in this application, the earlier the memory access is known, the earlier the instruction is fetched from memory and the earlier the instruction can be decompressed and dispatched. It would have been obvious to combine the continuation bit within the cache line of Miller because of the advantages provided above.

(10) Response to Argument

15. Regarding claims 1 and 8 rejected under 35 U.S.C. 102(b) as being anticipated by Miller et al. (U.S. 5,819,058):

Applicant argues "Miller failed to teach at least one instruction comprising information, inserted at compile time, that signals explicitly how the processing unit, when processing the instruction from a current memory line, should control how a part of processing is affected by crossing of a boundary to a subsequent memory line."

This argument is not found to be persuasive for the following reasons. Figures 3a and 3b show respectively 16-bit and 32-bit compressed instructions. Figure 4 shows how an uncompressed very long instruction word (VLIW) can be compressed down, which can be done by a compiler (Miller: Column 6 lines 56-58). Since the compression is performed by the compiler, then the end-of-packet (EP)/not-end-of-packet (NEP) field (element 117) of figure 3b must be inserted by the compiler. Thus, the compiler inserts information into at least one instruction.

The EP/NEP field, element 117 in figure 3b, determines if the current instruction is the last instruction within a compressed VLIW instruction (Miller: Column 6 lines 31-

35). Looking at figures 7 and 8, specifically instructions IP3 and IP4, both are compressed VLIW packets that cross into a subsequent memory line. Figure 8 shows how the fetching of the instructions occurs when this crossing of a boundary to a subsequent memory line works. A whole memory line can be fetched at a time when fetching instructions from memory (Miller: Column 8 lines 11-29 and column 11 lines 58-61), which means for IP3 and IP4, part of the subsequent memory line must also be fetched so that the whole VLIW packet can be processed (Miller: Column 12 lines 7-9 and lines 17-20).

Figures 7 and 8 are only an example of a scenario of fetching instructions from memory and the examiner will further assume that IP3 is made up of two 32-bit instructions for simplification. The examiner notes that IP3 could be made up of four 16-bit compressed instructions, but won't consider this since a different mechanism is used to indicate that a 16-bit instruction is the last instruction in a VLIW packet. Thus, in figure 8 IP3 consists of a first 32-bit compressed instruction in memory segments 6-7 and a second 32-bit compressed instruction in memory segments 8-9.

When processing IP3, the first 32-bit compressed instruction is processed. The first 32-bit compressed instruction will indicate in element 117 that the instruction is not at the end of the packet. Therefore, the next subsequent memory line must be read out of the memory to get the whole VLIW packet to completely decompress the VLIW packet (Miller: Column 12 lines 7-9). Thus, element 117 signals how to continue processing from a subsequent line by stating that the first compressed instruction isn't

the end of the VLIW packet and the end of the packet instruction must be fetched from the subsequent line.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

JAP

Jacob A. Petranek

April 23, 2007

Conferees:

Eddie Chan

Lynne Browne

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Lynne H. Browne
Appeal Specialist, TQAS
Technology Center 2100